

## Pengendalian Posisi *Mobile Robot* Menggunakan Metode *Neural Network* Dengan Umpan Balik Kamera Pemosisian Global

Randy Reza Kautsar, Bima Sena Bayu D, A.R. Anom Besari. S  
Program Studi Teknik Komputer

Politeknik Elektronika Negeri Surabaya – Institut Teknologi Sepuluh Nopember (ITS) Surabaya  
Kampus ITS, Sukolilo, Surabaya 60111

### Abstrak

*Pengembangan teknik otomasi pergerakan robot untuk dapat beroperasi di dunia nyata sudah menjadi bahan penelitian bagi pengembangan mobile robot di dunia saat ini. Untuk dapat mencapai suatu posisi yang diinginkan dari posisi awal, mobile robot membutuhkan suatu sistem navigasi yang dapat mengarahkan mobile robot tersebut ke posisi yang diinginkan. Untuk itu peneliti membuat sistem pengendali robot menggunakan metode neural network dengan menggunakan umpan balik kamera. Kamera berfungsi sebagai pemantau pergerakan robot yang nantinya akan diumpan balikkan ke dalam PC untuk mengetahui jalur yang harus ditempuh oleh robot dan posisi robot. Neural network digunakan untuk kontrol posisi pergerakan robot, fase learning neural network dilakukan secara offline (didalam PC) sedangkan fase mapping (penggunaan) dilakukan pada mikrokontroler yang terdapat pada robot. Data yang akan diajarkan dalam proses learning adalah data error sudut dan nilai PWM motor kanan dan PWM motor kiri. Persen keberhasilan dalam pengenalan data yang diajarkan adalah 95%. Sedangkan persen keberhasilan untuk pengontrolan robot menggunakan metode neural network backpropagation adalah 70%.*

Kata kunci: *mobile robot, neural network, control tracking, webcam*

### 1. PENDAHULUAN

Perkembangan dunia komputer, terutama dalam mikrokontroler memberikan dampak yang sangat besar terhadap dunia robotika, yang berkembang dengan begitu pesat. Hal ini tentu saja menuntut manusia untuk dapat mengikuti perkembangannya. Banyak pembahasan yang telah dilakukan dalam hal mesin atau alat yang mampu melakukan pembelajaran, dan bagaimana hal ini dapat diaplikasikan pada suatu *mobile robot* untuk membantu meningkatkan kemampuan operasional. *Mobile robot* merupakan konstruksi robot yang ciri khasnya adalah

mempunyai aktuator berupa roda untuk menggerakkan keseluruhan badan robot, sehingga robot tersebut dapat melakukan perpindahan posisi dari satu titik ke titik yang lain. Salah satu hal yang penting dalam perencanaan dan pembangunan pada sistem *mobile robot* adalah masalah olah gerak. Olah gerak yang dimaksud terdiri dari perencanaan gerak dan pengambilan keputusan pergerakan untuk menghindari terjadinya tabrakan dalam sebuah lingkungan.

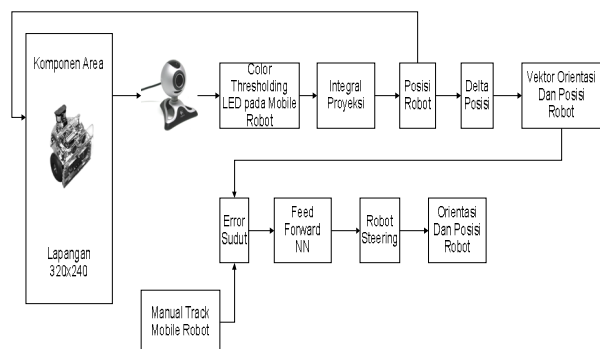
Pada penelitian ini akan dibuat suatu sistem pengendali *mobile robot* menggunakan metode *neural network* dengan menggunakan umpan balik kamera. *Neural network* merupakan suatu sistem yang terdiri dari arsitektur jaringan syaraf dan metode pembelajaran. *Neural network* secara umum digunakan untuk menirukan sistem kerja otak yang memiliki kemampuan untuk belajar (beradaptasi atau mengikuti perubahan dan belajar atau menerima sesuatu yang baru). Dengan menggunakan metode *neural network* ini *mobile robot* akan melakukan proses pembelajaran tertentu bagaimana untuk bergerak maju, mundur, ke kiri, ke kanan atau kemungkinan lain berdasarkan pengalaman error yang terjadi. Pada sistem ini fase pembelajaran *neural network* dilakukan secara offline (didalam PC) sedangkan fase *mapping* (penggunaan) dilakukan di dalam mikrokontroler yang terdapat pada robot.

Pada sistem ini juga digunakan sebuah kamera yang berfungsi sebagai pemantau pergerakan robot yang nantinya akan diumpan balikkan ke dalam PC untuk mengetahui apakah jalur yang dilalui robot sudah sesuai atau belum, jika tidak sesuai maka PC akan mengirimkan perintah ke mikrokontroler yang terdapat pada robot agar menggerakkan roda untuk bergerak ke kiri atau ke kanan sehingga robot dapat kembali lagi pada jalur yang sesuai. Sedangkan untuk pengiriman data dari PC ke mikrokontroler menggunakan komunikasi serial dengan media *Bluetooth*.

### 2. PERANCANGAN SISTEM

Secara umum proses perancangan sistem ini dibagi atas dua bagian utama yaitu bagian pertama yang membahas perancangan *software* pembentukan jalur secara manual di dalam PC dan *software* untuk mendeteksi

pergerakan robot. Pada bagian kedua akan membahas mengenai perancangan sistem pengendalian *mobile robot* yang dikerjakan di dalam mikrokontroler agar dapat bergerak mengikuti jalur secara optimal. Adapun blok diagram sistem secara keseluruhan terdapat pada gambar 1.



**Gambar 1.** Diagram blok sistem keseluruhan

Pengambilan gambar lapangan dan robot menggunakan kamera *webcam* dengan resolusi VGA 320x240 piksel. Posisi robot diidentifikasi dengan metode pengolahan citra, dengan menggunakan *color thresholding* dan integral proyeksi untuk mendapatkan informasi titik tengah dari robot.

Hasil dari identifikasi posisi robot akan dibandingkan dengan posisi jalur yang ditentukan apabila terjadi perbedaan maka akan terjadi error sudut. Error sudut ini akan menjadi input dari *neural network* dan akan menghasilkan output berupa PWM untuk menggerakkan roda pada robot untuk mengorientasi posisi robot sehingga robot bergerak sesuai dengan jalur yang ditentukan.

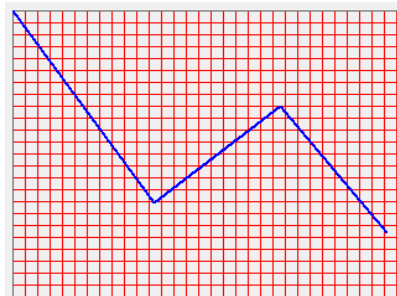
## 2.1. Pembuatan Software

### 2.1.1. Pembuatan Jalur *Mobile Robot*

Proses pembuatan jalur ini dilakukan pada layar *interface* yang terdapat di dalam PC dengan cara membuat beberapa titik yang masing-masing memiliki kordinat x dan y. Titik-titik ini akan dihubungkan sehingga menjadi sebuah garis yang mana garis inilah yang nantinya akan menjadi jalur yang akan dilalui oleh *mobile robot*. Gambar 2 menunjukkan hasil pembuatan jalur manual pada layar *interface*.

### 2.1.2. Identifikasi Posisi dan Pergerakan Robot

Identifikasi posisi robot menggunakan pengolahan citra untuk mendapatkan posisi robot dalam piksel. Proses yang digunakan dengan menggunakan *thresholding* untuk mendapatkan citra dari obyek tersebut. *Thresholding* yang digunakan menggunakan *distance thresholding*, pada



**Gambar 2.** Hasil pembuatan jalur manual pada layar *interface*

prinsipnya *distance thresholding* melakukan perhitungan perbedaan nilai warna RGB dengan nilai warna RGB acuan dari nilai RGB obyek. Perbedaan dihitung dengan *euclidean distance* menggunakan persamaan berikut

$$d = \sqrt{(R - r)^2 + (G - g)^2 + (B - b)^2} \quad \dots(1)$$

Dimana :

d = distance

R = nilai warna merah obyek

r = nilai warna merah *threshold*

G = nilai warna hijau obyek

g = nilai warna hijau *threshold*

B = nilai warna biru obyek

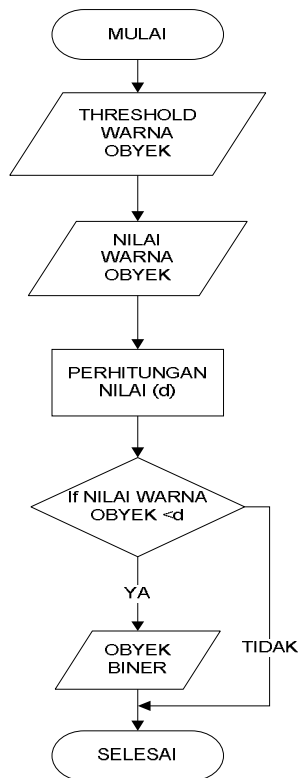
b = nilai warna biru *threshold*

*Threshold* digunakan untuk melakukan pembatasan nilai warna pada piksel sehingga dapat dikonversi menjadi gambar biner. Proses *distance thresholding* dapat dilihat pada gambar 3.

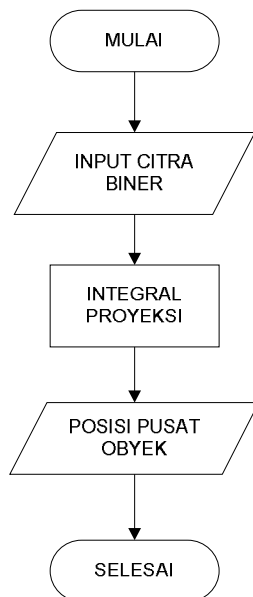
Setelah didapatkan gambar biner dari proses *threshold*, untuk mendapatkan nilai posisi x dan y pada piksel dilakukan integral proyeksi pada gambar biner. Proses integral proyeksi dapat dilihat pada gambar 4. Integral proyeksi melakukan *scanning* dari piksel minimum x dan minimum y sampai maksimum piksel x dan maksimum piksel y untuk mendapatkan nilai piksel dengan nilai 255 (jika obyek *threshold* dirubah dalam warna putih 255) atau 0 (jika obyek *threshold* dirubah dalam warna hitam). Setelah didapatkan nilai piksel minimum dimulainya gambar biner dan nilai maksimum gambar biner, maka untuk mendapatkan posisi titik tengah dari obyek dapat dihitung sesuai dengan persamaan berikut

$$pusat\_x = \frac{(\min\_x + \max\_x)}{2} \quad \dots(2)$$

$$pusat\_y = \frac{(\min\_y + \max\_y)}{2} \quad \dots(3)$$



Gambar 4 Flowchart distance threshold



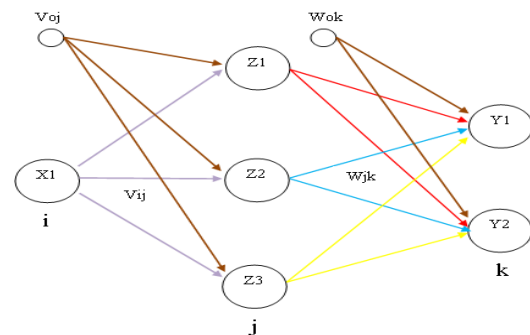
Gambar 5 Flowchart integral proyeksi

## 2.2. Sistem Kendali Mobile Robot

Pada proyek akhir ini, kendali pada *mobile robot* yang digunakan adalah sistem kendali Jaringan Saraf

Tiruan perambatan – balik (*JST backpropagation*). Dalam proses algoritma pelatihan Jaringan Saraf Tiruan perambatan – balik atau propagasi balik terdiri dari 3 tahapan yaitu :

- Pelatihan pola input secara umpan maju (*feed forward*).
- Perhitungan dan propagasi balik dari kumpulan kesalahan (error).
- Penyesuaian bobot atau tahap pengupdatean bobot.



Gambar 5 Arsitektur JST Backpropagation

Dari gambar di atas algoritma sistem pembelajaran jaringan saraf tiruan dapat diuraikan secara rinci sebagai berikut :

**Langkah 0** : Inisialisasi semua bobot dengan bilangan acak kecil

**Langkah 1**: Jika kondisi penghentian belum terpenuhi, lakukan langkah 2 sampai dengan 9

**Langkah 2**: Untuk setiap pasang data pelatihan, lakukan langkah 3-8

**Fase I** : Propagasi maju

**Langkah 3** : Unit input  $x_i$  ( $i=1,...,n$ ) menerima sinyal input  $x_i$  dan disebarkan ke semua unit *hidden layer*.

**Langkah 4** : Hitung semua output di *hidden layer*  $z_j$  ( $j=1,...,m$ ) dengan menjumlahkan faktor penimbang

$$Z\_in_j = V_{oj} + \sum_{i=1}^n x_i V_{i,j} \quad \dots(4)$$

Dan menghitung sesuai dengan fungsi aktivasi (*sigmoid*)

$$Z_j = f(Z\_in_j) = \frac{1}{1 + e^{-Z\_in_j}} \quad \dots(5)$$

Dan kemudian mengirim semua sinyal tersebut kesemua unit *output layer*

**Langkah 5** : Hitung semua output di *output layer*  $Y_k$  ( $k=1,...,p$ ) dengan menjumlahkan faktor penimbang

$$Y\_in_k = W_{ok} + \sum_{j=1}^m Out_j w_{j,k} \quad \dots(6)$$

Dan menghitung sesuai dengan fungsi aktivasi (*sigmoid*)

$$Y_k = f(Y\_in_k) = \frac{1}{1 + e^{-Y\_in_k}} \quad \dots(7)$$

### Fase II : Propagasi mundur

**Langkah 6** :Masing-masing unit output  $Y_k$  ( $k=1,...,p$ ) menerima pola target ( $t_k$ ) sesuai dengan pola masukan saat *training* dan menghitung nilai error.

$$\delta_k = (t_k - Y_k) f'(Y_{in_k}) = (t_k - Y_k) Y_k (1 - Y_k) \dots (8)$$

$\delta_k$  merupakan unit kasalahan yang akan dipakai dalam perubahan bobot layer hidden

Hitung suku perubahan bobot  $w_{kj}$  (yang akan dipakai nanti untuk mengubah bobot  $w_{jk}$ ) dengan laju pemahanan (*learning rate*)  $\alpha$ .

$$\Delta w_{jk} = \alpha \delta_k Z_j \dots (9)$$

**Langkah 7** :Masing-masing unit hidden  $Z_j$  ( $j=1...m$ ) menjumlahkan delta inputnya (dari *output layer*)

$$\delta_{in_j} = \sum_{k=1}^p \delta_k w_{jk} \dots (10)$$

Kemudian dikalikan dengan turunan dari fungsi *sigmoid*-nya untuk menghitung nilai error

$$\delta_j = \delta_{in_j} f'(Z_{in_j}) = \delta_{in_j} Z_j (1 - Z_j) \dots (11)$$

Hitung suku perubahan bobot  $w_{ij}$  (yang akan dipakai nanti untuk mengubah bobot  $w_{ij}$ ) dengan laju pemahanan (*learning rate*)  $\alpha$ .

$$\Delta w_{ij} = \alpha \delta_j x_i \dots (12)$$

### Fase III : Perubahan Bobot

**Langkah 8** :Hitung semua perubahan bobot Perubahan bobot garis yang menuju ke unit output

$$W_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk} \dots (13)$$

Perubahan bobot garis yang menuju ke unit hidden

$$W_{ij}(\text{baru}) = w_{ij}(\text{lama}) + \Delta w_{ij} \dots (14)$$

Setelah pelatihan selesai dilakukan, jaringan dapat dipakai untuk pengenalan pola. Dalam hal ini, hanya propagasi maju (langkah 4 dan 5) saja yang dipakai untuk menentukan keluaran jaringan.

Proses *learning backpropagation* dilakukan didalam PC. Pada tahap ini nilai awal dari bobot-bobot yang akan menjadi input dalam proses *learning backpropagation* merupakan nilai acak kecil (antara 0 - 1). Bobot-bobot ini akan diupdate secara terus-menerus dengan melihat selisih nilai output dan nilai target yang telah ditentukan. Proses ini akan terus dilakukan sampai mendapatkan nilai bobot yang optimal sehingga hasil outputnya mendekati nilai target.

Setelah mendapatkan bobot yang optimal, maka selanjutnya adalah melakukan proses *mapping*. Proses ini dilakukan di dalam mikrokontroller yang ada pada mobile robot. Dalam tahap ini nilai bobot-bobot yang optimal dijadikan input lagi dalam proses *learning backpropagation* hanya saja proses *learning* ini tidak dilakukan secara berulang-ulang hal ini dikarenakan bobot-bobot yang dipakai adalah bobot yang paling optimal sehingga dalam sekali proses fase propagasi maju nilai output yang dihasilkan sudah mendekati nilai target.

### 3. HASIL PENGUJIAN

Pada pengujian ini akan dilakukan penerapan algoritma *mapping neural network backpropagation* pada mikrokontroler dengan menggunakan node hidden 10. Berikut adalah tabel pasangan data pembelajarannya.

**Tabel 1** Pasangan data pembelajarannya

Error Sudut	PWM Kanan	PWM Kiri
20	38	20
15	38	22
10	38	25
5	38	27
0	35	35
-5	27	38
-10	25	38
-15	22	38
-20	20	38

Nilai error sudut pada tabel diatas akan menjadi input dalam proses *mapping*, sedangkan nilai PWM kanan dan PWM kiri merupakan nilai output target yang diinginkan, nilai output ini nantinya akan dibandingkan dengan nilai output yang didapat dari proses *mapping*. Berikut adalah yang menunjukkan nilai output dari proses *mapping*.

**Tabel 2** Data output proses *mapping*

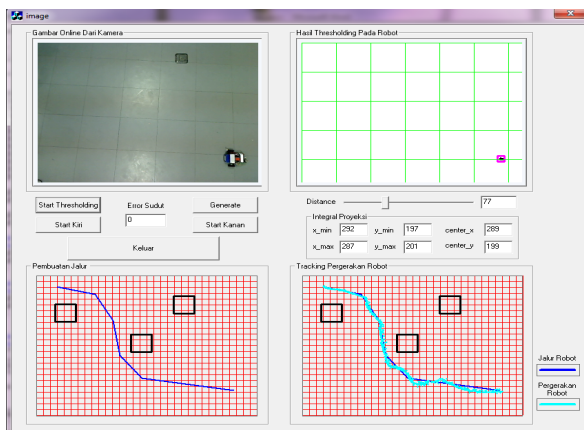
Error Sudut	Output Target		Output Proses	
	PWM Kanan	PWM Kiri	PWM Kanan	PWM Kiri
20	38	20	37	20
15	38	22	37	21
10	38	25	37	24
5	38	27	37	27
0	35	35	35	35
-5	27	38	26	37
-10	25	38	24	37
-15	22	38	21	37
-20	20	38	20	37

Dari hasil pengujian di atas terlihat bahwa nilai output proses *mapping* hampir mendekati dengan nilai output target sehingga persen error yang didapat untuk setiap nilainya tidak lebih dari 0,01%. Hal ini dikarenakan bobot yang digunakan pada proses *mapping* adalah bobot yang paling optimal yang didapat dari proses *learning* sehingga dalam sekali proses *forward backpropagation* nilai output yang dihasilkan dari proses *mapping* sudah

bisa mendekati nilai target. Gambar berikut merupakan pergerakan robot mengikuti jalur yang telah dibuat dengan menggunakan sistem *neural network backpropagation*. Rute jalur ditandai dengan garis biru tua sedangkan pergerakan robot ditandai dengan garis biru muda.

**Tabel 3** Data %error antara output proses *mapping* dengan output target

Error Sudut	% Error	
	PWM Kanan	PWM Kiri
20	0.01%	0%
15	0.01%	0.01%
10	0.01%	0.01%
5	0.01%	0%
0	0%	0%
-5	0.01%	0.01%
-10	0.01%	0.01%
-15	0.01%	0.01%
-20	0%	0.01%



**Gambar 4.30** Hasil pergerakan robot

Dari gambar di atas dapat dilihat bahwa pergerakan robot agak terlampaui jauh keluar lintasan namun robot berusaha bermanuver memutar untuk kembali kejalurnya lagi. Hal ini terjadi karena heading robot yang dideteksi kamera hanya satu. Sehingga pada saat robot bergerak melintasi jalur dengan posisi yang tidak sejajar dengan jalur error sudut yang dihasilkan sama dengan 0 sehingga robot berjalan lurus. Dengan demikian robot akan terlampaui keluar lintasan lagi sehingga berbalik kearah sebaliknya namun hal yang sama akan terjadi lagi robot kembali melintasi jalur dengan posisi yang belum sejajar dengan jalur sehingga robot kembali terlampaui keluar lintasan. Hal ini akan terus terjadi sampai posisi robot benar-benar sejajar dengan jalur.

#### 4. KESIMPULAN

Dari hasil pengujian yang telah dilakukan maka dapat diambil beberapa kesimpulan yaitu:

1. Faktor pencahayaan sangat berpengaruh pada proses *thresholding*.
2. Komunikasi serial dengan menggunakan *bluetooth* antara PC dan robot telah dapat dilakukan dengan baik.
3. Sistem kendali *neural network backpropagation* yang ditanamkan pada mikrokontroler robot sudah bekerja cukup bagus sesuai dengan yang direncanakan, dimana mampu memperkecil error orientasi sudut yang sudah ditentukan sebelumnya.
4. Program untuk mencari nilai error sudut antara jalur robot dan posisi robot dapat berjalan dengan baik.
5. 90%. Persen keberhasilan dalam pengenalan data yang diajarkan adalah 95%. Sedangkan persen keberhasilan untuk pengontrolan robot menggunakan metode *neural network backpropagation* adalah 70%.

#### DAFTAR PUSTAKA

- [1] Bima Sena Bayu Dewantara, "Perencanaan Jalur Mobile Secara Nyata Pada Lingkungan Dinamis Berbasis Compact Genetic Algoritma". Tesis Program Magister Teknik Elektro-Institut Teknologi Sepuluh Nopember. Surabaya. 2010.
- [2] Arie Setya Wulandari, Eru Puspita S.T., M.Kom "Model Pembelajaran Off-Line Menggunakan Jaringan Syaraf Tiruan Untuk Pengemudian Otomatis pada Kendaraan Beroda". Jurusan Teknik Elektronika PENS 2009.
- [3] Endah Sri Utami, "Pembacaan Plat Kendaraan Menggunakan Jaringan Saraf Tiruan (JST) Backpropagation Berbasis Image Processing", Proyek Akhir Politeknik Elektronika Negeri Surabaya, 2009.
- [4] Trendy Fibri Syamsiar, "Sistem Identifikasi Scan Iris Mata Menggunakan Metode JST Propagasi Balik untuk Aplikasi Sistem Pengamanan Brankas", Proyek Akhir Politeknik Elektronika Negeri Surabaya, 2009.
- [5] Bryson dan Ho, Paul Werbos, David E. Rumelhart, Geoffrey E. Hinton dan Richard J. Williams, "Metode Jaringan Saraf Tiruan: BACKPROPAGATION (PROPAGASI BALIK)", 1986